AMENDMENT TO THE SPECIFICATION:

Please amend the specification as follows:

[0001] This application claims priority under 35 U.S.C. § 119(e) to United States Provisional Patent Application No. 60/455,522, entitled "System and Method for Collecting, Modeling and Querying Information Technology Information" by Miller, et al., filed on March 18, 2003, which is hereby fully incorporated by reference herein. This application is related to United States Patent Application Serial No. 10/803,133, entitled "Method and System for Querying an Applied Data Model" by Miller et al., filed March 17, 2004 and United States Patent Application Serial No. 10/802,304, entitled "Method and System For a Generic Data Model" by Miller et al., filed March 17, 2004, each of which is fully incorporated by reference herein.

[0031] As an example, a particular computer in an IT environment can be represented with a component having attributes associated with the computer (e.g., amount of free memory, CPU and so on) and a software program can be represented by a component having attributes of the program. One or more relationship discovery rules can be applied to the computer component and software program component to determine if the computer has sufficient RAM to run the software program and an operating system ("OS") type that can support the software program. If the criteria of the rules are satisfied, a relationship can be established linking the software program component to the computer component, indicating that, in the IT environment being modeled, the computer system can support the software program. The relationship discovery rules applied to components can be arbitrarily complex, and can be applied globally or to a limited number of components. Application of a particular relationship discovery rule can bind zero components if the criteria of the rule are not satisfied, or two or more components if the criteria are satisfied.

[0032] According to one embodiment of the present invention, entities in a system and their interactions can be modeled using a generic data model such as that described in United States Patent Application Serial No. 10/802,304, entitled “Method and System For a Generic Data Model”, filed March 17, 2004, by Miller et al. (the “Generic Data Model Application”), which is fully incorporated by reference herein. Components for the generic data model can be established in any automated or manual manner as would be understood by those of ordinary skill in the art. Each component, when established, can be associated with one or more relationship discovery rules. The relationship discovery rules can be periodically applied by, for example, execution of a software program to determine if relationship between the associated component and one or more other components should be established or changed. If a relationship is established, it can be stored in a database table or other data storage format known in the art.

[0038] Additionally, the computer program can maintain relationship discovery rules to be applied to components to determine if a relationship exists. The relationship discovery rules, in one embodiment of the present invention, can be implemented as a script, such as a JAVA script, that can analyze particular components or existing relationships. As **[[and]]** an example, a relationship discovery rule 202 can be established for a “Runs On” relationship that specifies that if the following criteria are met a “Runs On” relationship should be established: the OS property in the two components match, the Free Space available in one component is greater than or equal to the footprint of the other component, and the “host name” value equals a “server name” value.

[0049] Additionally, component 300 can include a field for relationship discovery rules 350. The field, according to one embodiment of the present invention, can name a particular script or scripts that define the relationship discovery rules. The software program executing the relationship discovery rule can read the script name in field 355, find the named script and execute the named script. In an~~other~~ embodiment of the present invention, field 355 can contain the programming logic for the relationship discovery rule.

[0062] According to one embodiment of the present invention, a relationship discovery rule can be maintained as a script, such as JAVA script, that is executable to search and analyze data associated with components and/or relationships. Searches of the relationship and component data can be carried out, in one embodiment of the present invention, according to the query scheme described in United States Patent Application Serial No. 10/803,133, entitled "Method and System for Querying an Applied Data Model" filed March 17, 2004, by Miller et al., which is hereby fully incorporated by reference herein. A component can contain a reference to a script name or the script language itself. When a script referenced by a particular component is executed, the script can load specified values from the component by which the script is referenced and search for other components or relationships that satisfy the criteria of the script.

[0069] FIGURE 8 illustrates how relationships can change with changes of the IT system being modeled. Suppose that the IT environment depicted in FIGURE 7 is altered to the IT environment depicted in FIGURE 8. The IT environment now includes second server computer 140, with database program 132 executing on second server computer 140. Server computer 110 and web server program 130 can remain unaltered. The application of relationship discovery rules to the augmented IT environment 100 can, as will be described in conjunction with FIGURE 9, help identify a possible problem with the update to IT environment 100.

[0091] FIGURE 18 illustrates one possibility for an architecture to implement, store and access a data model, and the associated table schema and applied data models, implemented in accordance with described embodiments. System architecture 1800 consists of manager 1820, model 1830, database connectivity 1840 and database 1810. In one embodiment, database 1810 is configured to store the tables schema associated with the data model implemented by model 1830. Database 1810 may be a Microsoft SQL server, an Oracle database, IBM DB2, or the like. As will be readily apparent to those of ordinary skill in the art, many sorts of databases may be used to implement these table schemas.